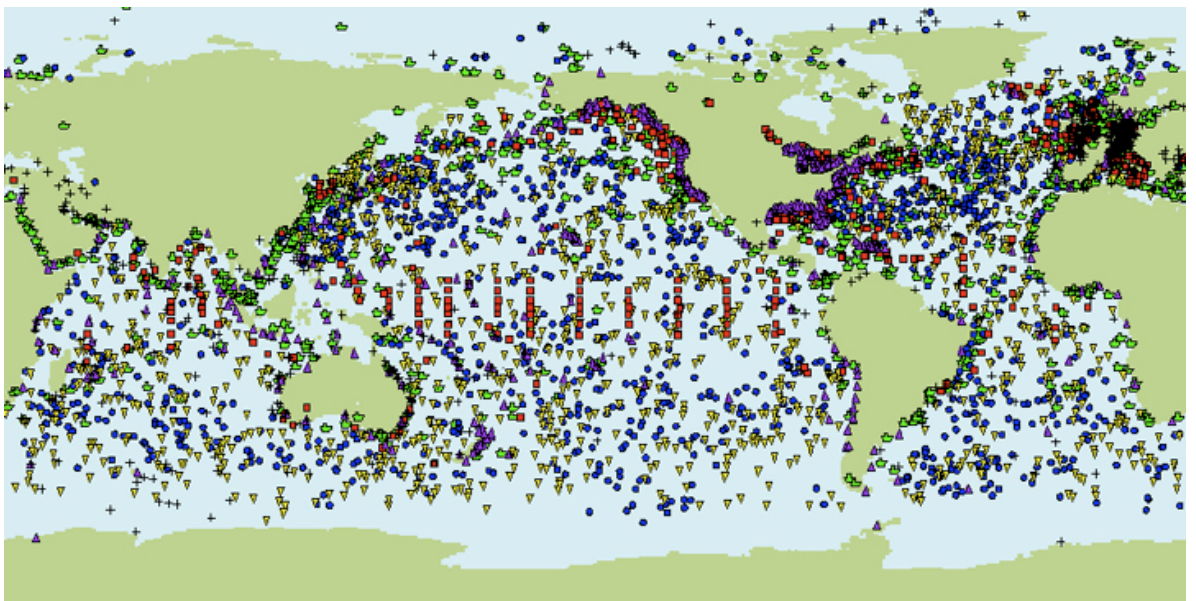
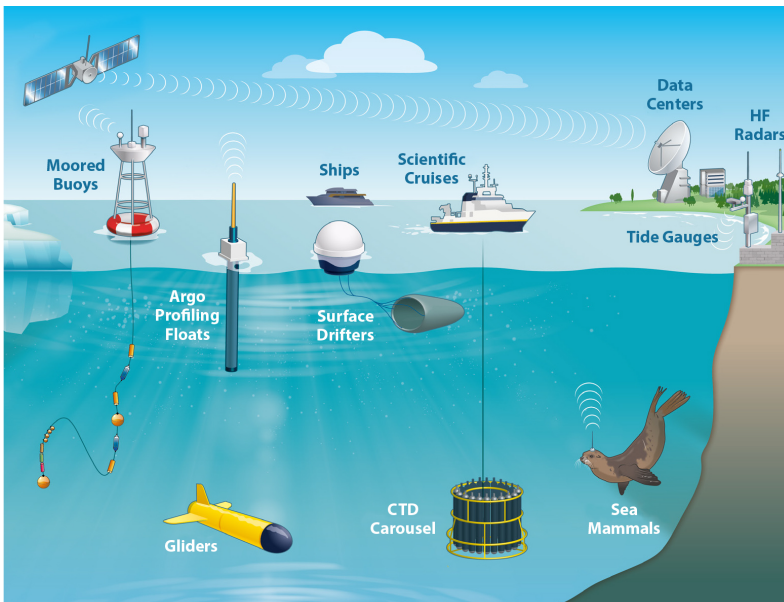


## ✓ Python을 이용한 해양데이터 활용 실습

### 학습목표

1. 해양관측데이터란 무엇이고 수집하는 방법을 알 수 있다.
2. python을 이용하여 우리나라 해저지형도를 만들 수 있다.
3. OpenAPI를 이용하여 실시간 해양정보를 활용할 수 있다.

## ✓ 해양관측데이터의 개요와 대표적인 웹사이트



- WOD
- BODC
- Copernicus

#### 국내

- 국립해양조사원
- 기상청
- 국립수산물과학원
- 해양환경공단
- JOISS

## Markdwon

#### 참고자료

- [https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat\\_sheet\\_for\\_Google\\_Colab.ipynb](https://colab.research.google.com/github/Tanu-N-Prabhu/Python/blob/master/Cheat_sheet_for_Google_Colab.ipynb)
- [https://colab.research.google.com/notebooks/markdown\\_guide.ipynb](https://colab.research.google.com/notebooks/markdown_guide.ipynb)

## Google Earth Engine 소개

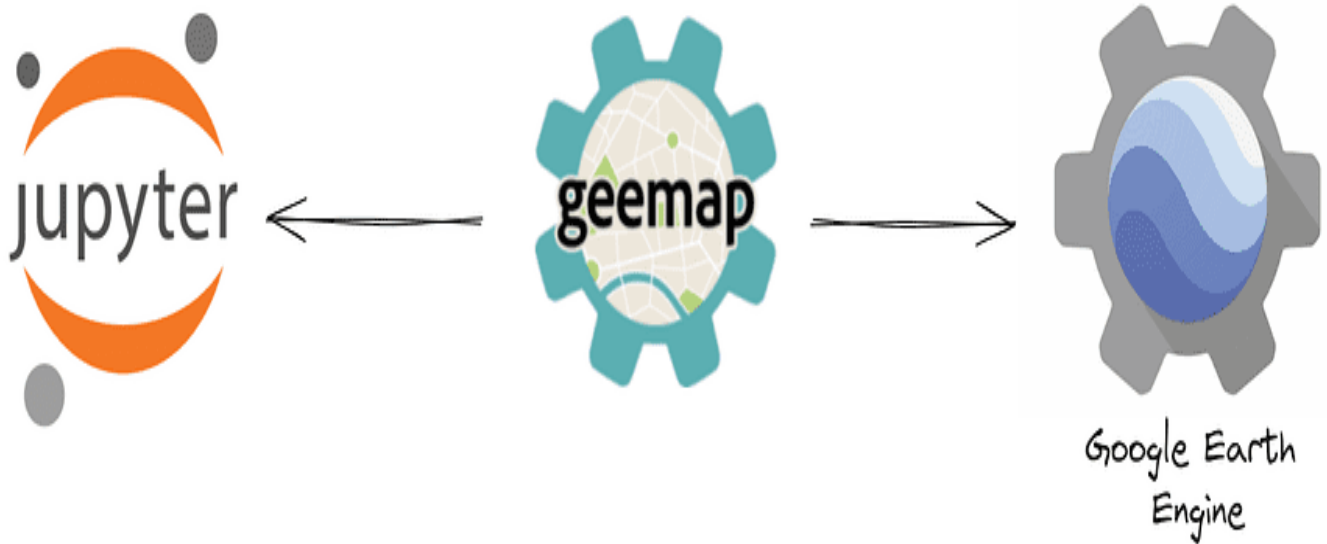
GEE는 지리공간데이터 분석 및 시각화 플랫폼으로써 위성영상 및 40년 이상의 지구관측 이미지 데이터와 함께 학술, 영리 및 비영리, 공공에 서비스

- 개방 데이터 카탈로그, 병렬계산 컴퓨팅 인프라, 지리공간API, 인터랙티브 앱 서버 제공

### Google Earth와 무엇이 다른가?

- 구글어스는 가상지구본을 통한 세계 탐색 도구인 반면, 어스엔진은 지리공간데이터 분석 도구

## GEEMAP



#### 참고문헌

Wu, Q., (2020). geemap: A Python package for interactive mapping with Google Earth Engine. The Journal of Open Source Software, 5(51), 2305. <https://doi.org/10.21105/joss.02305>

## ✓ Earth Engine Python API Colab Setup

### ✓ API 불러오기(import)와 토큰 인증

EE API는 Colab에 기본 설치되므로 불러오기와 인증단계만을 요구함. Colab 커널을 재시작하거나 Colab 가 상머신이 비활성화로 인해 재시작하는 경우에도 이 단계를 거쳐야 함.

### ✓ API 불러오기

```
import ee
```

### ✓ Authenticate and initialize

- `ee.Authenticate` 함수 실행: Earth Engine 서버 접속을 승인하기 위함
- `ee.Initialize` : 초기화
- 이 단계를 거치면 Earth Engine에 Google 계정 액세스 권한을 부여하라는 메시지가 표시되며 셀에 인쇄된 지침을 따름

```
# Trigger the authentication flow.
ee.Authenticate()

# Initialize the library.
ee.Initialize(project='my-project')
```

## ✓ GEE 기초

- GEE 둘러보기
  - GEE Code Editor
  - GEE Data Catalog
- GEE 객체
  - ee.Image
  - ee.ImageCollection
  - ee.Feature
  - ee.FeatureCollection
  - ee.Geometry
- 실습: 한반도 상공의 Landsat 위성이미지 찾기

## ✓ Test the API

Test the API by printing the elevation of Mount Everest.

```
# Print the elevation of Mount Everest.
dem = ee.Image('USGS/SRTMGL1_003')
xy = ee.Geometry.Point([86.9250, 27.9881])
elev = dem.sample(xy, 30).first().get('elevation').getInfo()
print('Mount Everest elevation (m):', elev)
```

⇒ Mount Everest elevation (m): 8729

## ✓ Map visualization

### ✓ Interactive map

[geemap](#) 라이브러리를 사용하여 ee.Image 객체를 대화형 [ipyleaflet](#) 맵에 표시해자.

```
%pip install -U geemap
```

⇒ [Show hidden output](#)

```

import geemap

roi = ee.Geometry.Point(128.9697, 35.1796)

dataset = ee.ImageCollection('LANDSAT/LC09/C02/T1_L2').filterDate(
    '2022-01-01', '2022-12-01').filterBounds(roi).filter(ee.Filter.lt('CLOUD_COVE

# Applies scaling factors.
def apply_scale_factors(image):
    optical_bands = image.select('SR_B.').multiply(0.0000275).add(-0.2)
    thermal_bands = image.select('ST_B.*').multiply(0.00341802).add(149.0)
    return image.addBands(optical_bands, None, True).addBands(
        thermal_bands, None, True
    )

dataset = dataset.map(apply_scale_factors).first()

visualization = {
    'bands': ['SR_B4', 'SR_B3', 'SR_B2'],
    'min': 0.0,
    'max': 0.3,
}

m = geemap.Map()
m.centerObject(roi, 8)
m.add_layer(dataset, visualization, 'True Color (432)')
m

```



+

-

50 km

30 mi

All layers on/off

☒ OpenStreetMap....

☒ True Color (432)

ipyleaflet | © [OpenStreetMap](#) contributors, Google Earth Engine

```
# Import the geemap library.
import geemap
import geemap.colormaps as cm

eez = ee.FeatureCollection("projects/ee-haebom/assets/kr_eez")
gebco_grid = ee.ImageCollection("projects/sat-io/open-datasets/gebco/gebco_grid")
gebco_grid_ocean = gebco_grid.median().updateMask(gebco_grid.median().lt(0))

dataset = ee.Image('NOAA/NGDC/ETOP01');
elevation = dataset.select('bedrock');
elevation = elevation.updateMask(elevation.lt(0));

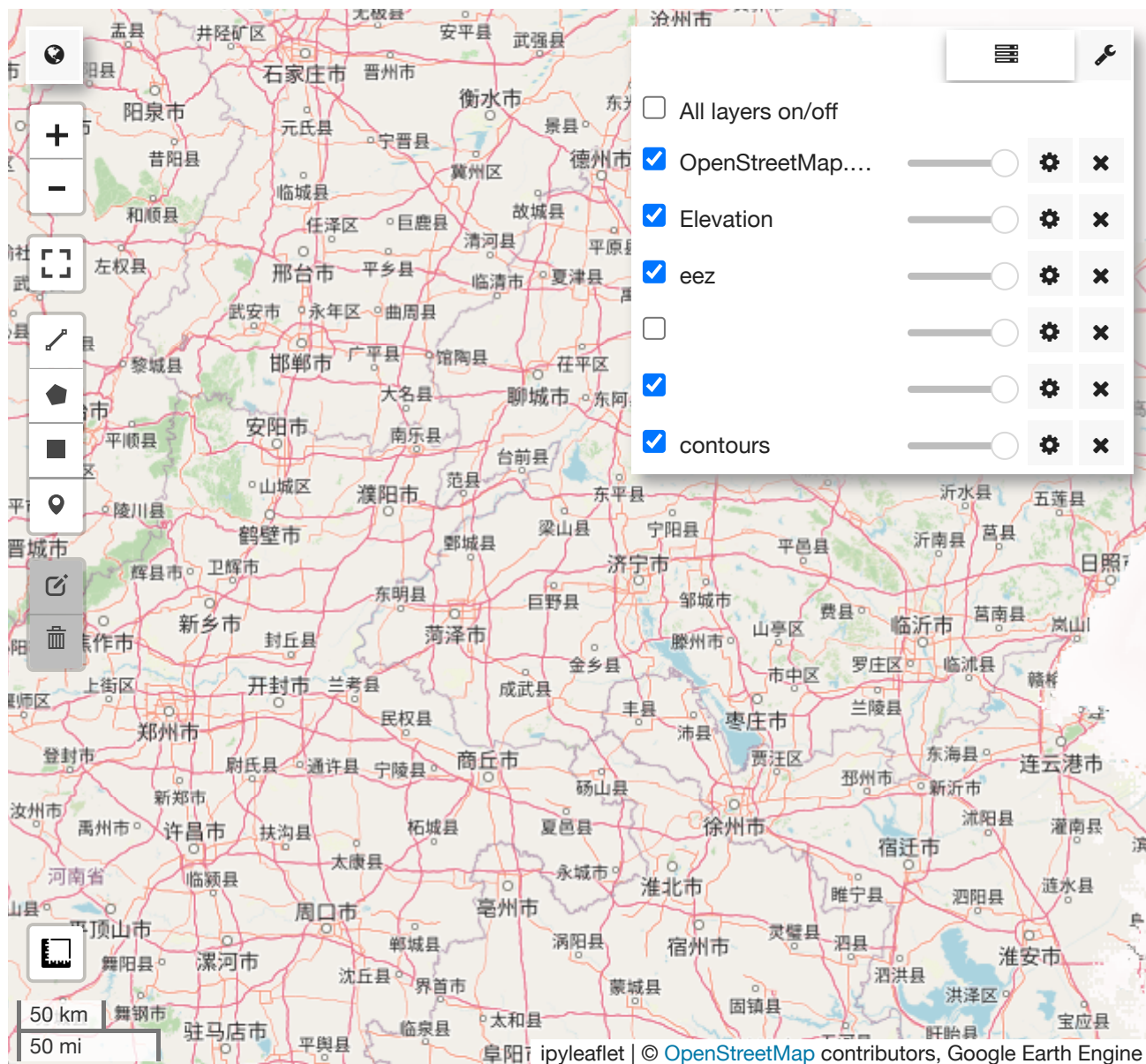
# Set visualization parameters.
elevationVis = {
    'min': -7000.0,
    'max': 0.0,
    'palette': cm.palettes.gist_earth }

#["#4C00FFFF", "#0019FFFF", "#0080FFFF", "#00E5FFFF", "#00FF4DFF", "#4DFF00FF", "#"]

# Create a map object.
Map = geemap.Map(center=[37, 129], zoom=7)

# Add the elevation model to the map object.
Map.addLayer(elevation, elevationVis, 'Elevation')
Map.addLayer(eez, {}, 'eez')

# Display the map.
display(Map)
```



# 등수심선(contour lines) 생성 함수

```
def contour_f(line):  
    binary_contour = elevation.convolve(ee.Kernel.gaussian(7, 5)).subtract(ee.Image.constant(0)).mask(binary_c  
    return binary_contour.multiply(ee.Image.constant(line)).toFloat().mask(binary_c
```

```
lines = ee.List.sequence(-4000, 0, 40)  
contourlines = lines.map(contour_f)  
contourlines = ee.ImageCollection(contourlines).mosaic()  
contourlines = contourlines.clip(eez);
```

# Add a layer to show the contour lines

```
Map.addLayer(contourlines, {'min': -4000, 'max': 0, 'palette': cm.palettes.gist_e
```





# 우리나라 관할해역(EEZ)에서 가장 깊은 곳의 좌표와 수심을 찾아보자

- Reducer 사용 : Reducer는 어스엔진에서 시간, 공간, 밴드, 배열과 같이 각각의 구조화된 데이터를 집계하는 함수

- reduceRegion()

```
lonLatImage = ee.Image.pixelLonLat();
imageWithLonLat = elevation.addBands(lonLatImage);

stats = imageWithLonLat.reduceRegion(
  reducer=ee.Reducer.min(3),
  geometry=eez,
  scale=1000,
  crs= 'EPSG:4326', # California Albers projection
)

print(stats.getInfo())

df = stats.getInfo()['min']
x = stats.getInfo()['min1']
y = stats.getInfo()['min2']
Map.add_marker(
  [y, x],
  shape="circle",
  radius=20,
  color="red",
  fill_color="#3388ff",
  fill_opacity=0.5,
)
Map
```

➡ {'min': -3427, 'min1': 131.11360729366476, 'min2': 38.81171185038393}

